

PATENT
IBM Docket No. GB9-2000-0033

REMARKS

Status:

Claims 1 - 17 stand rejected under 35 U.S.C. §112 for indefiniteness. Claims 1-17 also stand rejected under 35 USC 103(a) as being unpatentable in view of the teaching of US Pat. No. 6,058,389 to Chandra et al., considered in view of the teaching of US Pat. No. 5,857,180 to Hallmark et al.

Claims 1-11, 13 and 15-16 are presented for reconsideration as explained in the discussion below. Claims 12, 14 and 17 are canceled.

Analysis:

The claims presented for reconsideration have been amended to clarify the invention and overcome the indefiniteness rejection. The term PUT, which was objected to as indefinite, is a term of art and an Internet dictionary page indicating the definition is attached (Appendix A) to indicate the meaning - to copy a file from a local computer to a remote computer. This meaning is reflected in lines 3-4 of claim 1.

In response to the Examiner's question, the cause of the commit is not considered part of the invention. It is an operation by the sender application that indicates its work is complete and other applications may use the message (see definition at Appendix B hereto).

As regards claim 13, the "postponing" is of applying an index key that has an attribute value specified by the sending application when the message was placed on the queue. This is referred to as the secondary index key in the Application (see p. 20, full

PATENT
IBM Docket No. GB9-2000-0033

paragraph). The claim is directed to that index and not to the primary index; which, as the Examiner suggested, does not align with the claim.

The term "other criteria" was questioned by the Examiner. It now has basis in the preamble and applies to additional criteria that may cause a message that matches regarding request attribute value to be rejected by the receiving program; such as entry date, or excluding data (wrong gender, wrong banking institution)) contained in the message.

The claims have been amended to emphasize the match of a request attribute value to an attribute value of an index key assigned upon or in response to commit of the associated message. Also the modifier terms "sender", for the application placing the message, and "receiver", for the application seeking a match, have been carried through to overcome antecedent basis problems identified by the Examiner.

As regards the Chandra teaching, it is asserted in the Office action that it covers "assigning an index key to a message in response to such commit of the put operation...". But Chandra teaches a basic database and does appear to even teach use of an index or an index key. At Chandra col.10, lines 3 and 4 the possibility of using indexes, somehow, is briefly mentioned but no teaching follows. We are left knowing only that SQL supports indexes; but, without a teaching of use for matching receiver requests with sender attribute values, let alone delayed assignment respective of message commit.

Looking further to the Chandra teaching, at Fig. 6A, commit is either assumed (box 604) or occurs at box 610. At Chandra col. 14, lines 4-7 it is indicated: "In step 610 the

PATENT
IBM Docket No. GB9-2000-0033

process calls another kernel function of the database server to commit the recursive transaction." But the database server is not described! It appears, the Chandra embodiment is based on an SQL interpreter (Chandra col. 9 line 56). Looking to Appendix B (attached hereto) a definition of "commit" in the SQL environment (def. 4) indicates: "*When a commit operation occurs, the locks are released to allow other applications to use the changed data.*" Where in Chandra is there a teaching that Fig. 6A step 610 does other than set and release locks as Applicants have indicated is prior art (Application p. 6 bottom): "*A known solution to this problem is to apply locks to each message while the message Put operation is in doubt.*"

Indexes are separately searchable See Applicants' specification at p. 9, last paragraph: "*The invention provides advantages over merely setting a commit flag on a message, since the index flag which is assigned at commit time is itself a searchable index...*". By assigning an attribute value at commit, a match to a request attribute value occurs only for committed messages - thus using the search facility to perform, in effect, the lock function. This avoids a subsequent, separate test of a lock flag for messages that match. This is a major time saver for a busy shared queue.

In accordance with the above, Applicants see Chandra as lacking a teaching for assigning attribute values from a sender application to an index key, much less, an assignment in response to commit of a PUT operation. Accordingly withdrawal of the rejections based on Chandra as a primary reference is respectfully solicited.

The other prior art including the Hallmark teaching does not make up for the fundamental deficiencies of Chandra explained above. Again, where is the teaching of assigning an index key with attribute values related to a message? And, where is the

PATENT
IBM Docket No. GB9-2000-0033

teaching to delay assignment until a commit of the message occurs? Where is the matching to the index key? The only teaching that appears to be known in the art is a separate lock/unlock process which adds another layer of complexity.

As indicated above, it is believed the claims now clearly identify Applicant's advance over the prior art. A clever way of setting up an index with special timing that eliminates the need for a lock/unlock capability to avoid premature access to messages on a shared queue. An approach not found in the prior art. Accordingly, early notice to the effect that this case has been placed in condition for allowance is earnestly solicited.

Applicant's attorney would welcome a telephone communication from the Examiner (to the telephone number indicated below), for purposes of advancing the prosecution of this case

Respectfully Submitted,


George E. Grosser

Reg. No. 25,6291

c/o IBM Corp.
Dept. T81/Bldg. 503 PO Box 12195
Research Triangle Park, NC 27709

Tel. No. (919)968-7847 Fax 919-254-4330
EMAIL: gegch@prodigy.net